A Holistic Multimedia System for Gastrointestinal Tract Disease Detection

Konstantin Pogorelov
Simula Research Laboratory, Norway
University of Oslo, Norway

Sigrun Losada Eskeland
Bærum Hospital, Norway

Thomas de Lange
Bærum Hospital, Norway
Cancer Registry of Norway

Carsten Griwodz
Simula Research Laboratory, Norway
University of Oslo, Norway

Kristin Ranheim Randel
Cancer Registry of Norway
University of Oslo, Norway

Håkon Kvale Stensland
Simula Research Laboratory, Norway
University of Oslo, Norway

Duc-Tien Dang-Nguyen
Dublin City University, Ireland

Concetto Spampinato
University of Catania, Italy

Dag Johansen
UiT–The Arctic University of Norway

Michael Riegler
Simula Research Laboratory, Norway
University of Oslo, Norway

Pål Halvorsen
Simula Research Laboratory, Norway
University of Oslo, Norway

ABSTRACT

Analysis of medical videos for detection of abnormalities and diseases requires both high precision and recall, but also real-time processing for live feedback and scalability for massive screening of entire populations. Existing work on this field does not provide the necessary combination of retrieval accuracy and performance.

In this paper, a multimedia system is presented where the aim is to tackle automatic analysis of videos from the human gastrointestinal (GI) tract. The system includes the whole pipeline from data collection, processing and analysis, to visualization. The system combines filters using machine learning, image recognition and extraction of global and local image features. Furthermore, it is built in a modular way so that it can easily be extended. At the same time, it is developed for efficient processing in order to provide real-time feedback to the doctors. Our experimental evaluation proves that our system has detection and localisation accuracy at least as good as existing systems for polyp detection, it is capable of detecting a wider range of diseases, it can analyze video in real-time, and it has a low resource consumption for scalability.

CCS CONCEPTS

• **Information systems → Multimedia information systems**;

KEYWORDS

Interactive; Medicine; Gastrointestinal Tract; Medical Multimedia System; Performance; Evaluation

This work is founded by the Norwegian FRINATEK project "EONS" (#231687).
Contact author's address: Konstantin Pogorelov, Simula Research Laboratory, Oslo, Norway, email: konstantin@simula.no .

MMSys '17, June 20–23, 2017, Taipei, Taiwan

ACM ISBN 978-1-4503-5002-0/17/06.
https://doi.org/http://dx.doi.org/10.1145/3083187.3083189

ACM Reference format:
Konstantin Pogorelov, Sigrun Losada Eskeland, Thomas de Lange, Carsten Griwodz, Kristin Ranheim Randel, Håkon Kvale Stensland, Duc-Tien Dang-Nguyen, Concetto Spampinato, Dag Johansen, Michael Riegler, and Pål Halvorsen. 2017. A Holistic Multimedia System for Gastrointestinal Tract Disease Detection. In *Proceedings of MMSys '17, Taipei, Taiwan, June 20–23, 2017,* 12 pages.
https://doi.org/http://dx.doi.org/10.1145/3083187.3083189

1 INTRODUCTION

The human gastrointestinal (GI) tract can potentially be affected by various abnormalities and diseases, including colorectal cancer (CRC) which is a major health issue world wide. For the case of CRC, an early detection is crucial for survival, and several studies demonstrate that a population-wide screening program improves the prognosis and even reduce the incidence of CRC [23]. As a consequence, in the current European Union guidelines, screening for CRC is recommended for the population over 50 years of age [57].

Colonoscopy, a common medical examination and the gold standard for visualizing the mucosa and the lumen of the entire colon, may be used either as a primary screening tool or as a work-up tool after other positive screening tests [33]. However, endoscopies are invasive procedures and may be of great discomfort for patients. Long-lasting training of physicians or nurses is required to perform the examinations. They are performed in real-time and are challenging to scale to a larger population. Additionally, the procedure is expensive. In the US, for example, the colonoscopy is the most expensive cancer screening process with annual costs of 10 billion dollars ($1100/person) [55], and with a time consumption of about one medical-doctor-hour and two nurse-hours, per examination.

In this respect, we propose a scalable, real-time disease-detection system for the GI tract. The idea is to assist endoscopists (physicians highly trained in the procedure) during live examinations. Additionally, alternatives to traditional endoscopy examinations have recently emerged with the development of non-invasive endoscopy capsules (WVCs). The idea is a pill-sized camera (available from vendors such as Given and Olympus), that is swallowed and

then records a video of the entire GI tract. The challenge in this context today is that medical experts still need to view the video in a non-scalable way. Our system should provide a scalable system that can be used as a first-order population screening system where the WVC-recorded video is used to determine whether a traditional endoscopic examination is needed or not.

The system presented in this paper is designed to support detection of a wide range of diseases, but our initial focus is on colorectal polyps and a small subset of other diseases. Polyps are specifically relevant because they are known precursors of CRC (see for example figure 2 and 3). The reason for starting with this scenario is that most colon cancers arise from benign, adenomatous polyps containing dysplastic cells that may progress to cancer. Detection and removal of such polyps prevents the development of cancer. Thus, the risk of getting CRC the following 60 months after a colonoscopy depends largely on the endoscopists ability to detect polyps [25].

In the context of object or pattern detection and tracking in images and videos, there has been a lot of research, and current systems are good at detecting human faces, cars, logos, etc. However, detecting diseases in the GI tract is very different from detecting objects like logos or cars. The GI tract can potentially have a wide range of lesions visible on endoscopy, as well as findings associated with benign/normal or man-made lesions. This leads to necessity of distinguishing between multiple classes of diseases, including findings with high level of visual similarity. In this scenario, both high precision and recall are of crucial importance, but also is the often ignored system performance in order to provide live feedback because medical personal is assisted most efficiently while they perform the examination. The most recent and most complete related work is the polyp detection system Polyp-Alert [61], which can provide near real-time feedback during colonoscopies. However, it is limited to polyp detection, and it is not fast enough in the case of live examinations.

To further aid and scale such examinations, we have developed EIR [47], an efficient and scalable information retrieval system for medical data like videos and images. The system supports endoscopists in the detection and interpretation of diseases in the GI tract. In this paper, we provide more detailed description of our EIR system, we greatly extend the evaluation, and we also introduce localization. The main objective of the system is to develop both (i) a live-system assisting the visual detection of diseases during colonoscopies, and (ii) a future fully automated first line screening for CRC using WVCs. Both goals pose strict requirements for the accuracy of the detection in order to avoid false negative findings (overlooking a disease) as well as low resource consumption. The live assisted system also introduces a real-time processing requirement (defined as being able to at least process 25 frames or images per second). In this paper, the initial prototype of our system is presented. This is built by combining filters using machine learning, image recognition and extraction and comparison of global and local image features. The system will be extended to support detection of multiple abnormalities and diseases of the GI tract by training the classifiers using different datasets. We evaluate our prototype by training classifiers that are based on the different image recognition approaches. It is important to point out that these classifiers can also process other input like for example sensor data. We also test the generated classifiers with different diseases and thereby evaluate the different approaches for feasibility of colonic polyp recognition and localisation.

The initial results from our experimental evaluation show that: (i) the detection and localisation accuracy can reach the same performance or outperform other current state of the art methods, (ii) the system performance reaches real-time in terms of video processing up to high definition resolutions.

The rest of the paper is organized as follows: we present related work in section 2. This is followed by a description of the complete system in section 3. After that, we present a detailed evaluation of the whole system in section 4, and we discuss in section 5 two cases where our system will be used in two medical examinations. Finally, we draw the conclusion in section 6.

2 RELATED WORK

To the best of our knowledge, no related work that presents a complete multimedia system for analysing the whole GI tract in real-time exists. The complete system covers the entire pipeline from data capture to live detection feedback, and has to fulfill many requirements. These requirements include (i) high detection accuracy, (ii) real-time processing to support live examinations like colonoscopies, (iii) efficient resource utilization to allow massive scale using WVCs, and (iv) expandability to allow the system to support new diseases.

Detection of diseases in the GI tract has mostly focused on polyps. This is most probably due to the lack of data in the medical field and polyps being a condition with at least some data available [30]. Automatic analysis of polyps in colonoscopies has been in the focus of researchers for a long time and several studies have been published [58, 59, 62]. However, not many systems are able to do real-time detection or support doctors by computer aided diagnosis during colonoscopies in real-time. Furthermore, all of them are limited to a very specific use case, which in the most cases is polyp detection for a specific type of camera.

Several algorithms, methods and partial systems have been proposed and have achieved results in their respective testing environment that are promising. However, most of the research conducted in this field uses a rather small amount of training and testing data, making it difficult to generalize the methods beyond the specific dataset and test scenarios. In the [47] paper, we presented a summary of the detection performance and speed properties of the most relevant approaches in colonoscopy and polyp detection. The conducted search through the relevant publications [3, 4, 9, 24, 26, 28, 34, 60, 61, 63] showed that different researchers provide different metrics for measuring the performance and use different datasets for training and testing. Moreover, almost all of the researches focus on polyps only.

The Polyp-Alert approach from Wang et al. [61] is the most recent, most complete and best working in the field of polyp detection. It is able to give near real-time feedback during colonoscopies. The system can process 10 frames per second and uses visual features and a rule-based classifier to detect the edges of polyps. Further, they distinguish between clear frames and polyp frames in their detection. The researchers report a performance of 97.7% correctly detected polyps, based on their dataset which consists of 52 videos

taken from different colonoscopes. Unfortunately, the dataset is not publicly available and therefore a detection performance comparison is not possible.

Mamonov et al. [34] presented an algorithm for a binary classifier with pre-selection to detect polyps in the colon. The used assumption is that polyps can be generalized as protrusions (something that bumps out) that are mostly round in shape. The researchers report a sensitivity of 81.25% per polyp at a specificity of 90%. The sensitivity of the algorithm with regards to single input frames is significantly lower and only reaches 47%. The length of an input sequence varied between 2 and 32 frames and a total of 16 sequences were tested. The false positive rate on the total of 18, 738 frames not containing a polyp was 9.8%. Assuming that it is usual to have multiple frames available for a single polyp, these numbers seem quite promising. With this method, the time a specialist has to spend on evaluating video data could be reduced by about 90%.

A similar approach is presented by Hwang et al. [24]. This approach also focuses on shape, in particular on ellipses, which is a common shape for a polyp. Using this method, a frame is first segmented into elliptical regions by a watershed-based image segmentation algorithm. These regions and corresponding ellipse edges are then evaluated for matching of curve direction, curvature, edge distance and intensity. After the first frame a potential polyp was detected, subsequent frames are also searched for the same characteristics using a mutual and information based image registration technique. To evaluate the method, a video sequence with a frame rate of 15 fps has been processed. Out of 27 available polyp shots (frames containing a polyp), 26 were detected correctly with a total of 5 false-positives. Similar to [34], the authors assume that multiple frames are available for one polyp and that a certain number of false-negatives is acceptable in order to balance the number of false-positives. The correctness of this assumption depends strongly on the frame rate of the camera that is used for recording the video.

Another recent approach related to our approach and not limited to polyps is presented by Nawarathna et al. [39]. In the paper, the authors describe a method to detect abnormalities like bleeding, but also polyps in colonoscopy videos. The authors use a texton histogram of an image block. The authors report a 91% recall and a 90.8% specificity for colonoscopy images.

Other papers that discuss how to improve performance of endoscopic surgeries in general (not colonoscopy) are for example [36–38]. In these papers, the authors report their method for detecting the circular content area that is typical in endoscopic videos. Furthermore, they present their method for relevance segmentation in endoscopic videos. The methods seem to be very useful in terms of archiving and saving storage space.

Since neural networks (NNs) are commonly used nowadays, they are also discussed for automatic analysis of GI tract videos. NNs are conceptually easy to understand and lately large amounts of academic research has been done on them. Results recently reported on, for example, the ImageNet dataset look quite promising [13]. Nevertheless, they have some negative aspects that make them less useful for our use case [10]. First, NNs are a *blackbox* approach. This can lead to serious problems in the medical field since it is not possible to evaluate them properly, and there will always be a chance that they completely fail without being aware of it [40]. Further, training of NNs is complicated, takes a long time and requires a lot of training data. In the medical field, this can be a challenge since it is hard to get data due to the lack of experts' time and because of legal and ethical issues. Some common conditions such as colon polyps may reach the required amount of training data for a NNs while other findings, like tattoos from previous endoscopic procedures, are not that well documented but still interesting to detect [48]. Finally, NNs are not easy to design for probabilistic results. In a multi-class decision-based system that is built to support medical doctors in decision making, the probability is an important information to help them finding a decision. Approaches with a better understanding of the problem give a much more accurate probabilistic score that can be directly translated to the real world scenario [50].

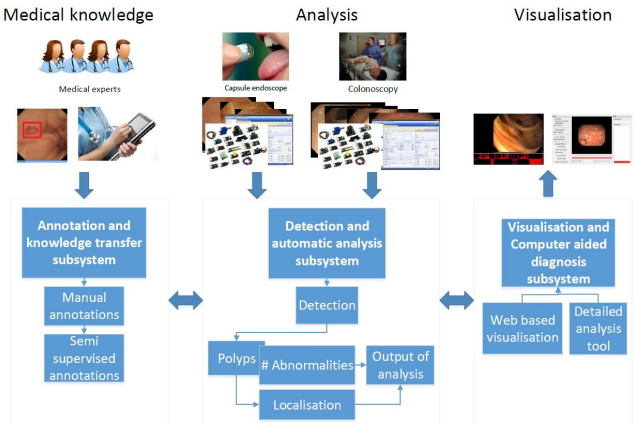


Figure 1: System overview with the three main subsystems: annotation, detection and automatic analysis and visualization.

In summary, a lot of related work with many interesting approaches for polyp detection exists. However, they (i) are either too narrow for a flexible, multi-disease detection system, (ii) have been tested on a too limited datasets not showing if the methods would work in a real scenario, or (iii) provide a too low performance for a real-time system or authors have ignored the system performance aspect in their evaluations altogether. To the best of our knowledge, our system is the first that aims at total flexibility in terms of diseases that can be detected, and at the same, time focuses on the performance and the evaluation of it.

3 BASIC IDEA OF THE SYSTEM

The objective of the system is to support doctors in GI tract disease detection, both as a live examination system and as an offline system for WVCs. Its main requirements are already listed in section 2, but it also has to be easy to use. Figure 1 gives an overview of the whole system. It consists of three main parts: the annotation subsystem, the detection and automatic analysis subsystem and the visualization and computer aided diagnosis subsystem.

3.1 Annotation Subsystem

It is well known that training data is very important for a classification system that relies on machine learning techniques. In the

medical field, both the time of the experts and available data are very limited. Even when experts' time can be acquired, the quality of annotations depends on their experience and concentration [17]. For each image or video, a patient consent has to be collected before research can be done, making it a very cumbersome task. The purpose of the annotation subsystem is therefore to efficiently collect training data for the detection and automatic analysis subsystem.

For example, in a single WVC procedure, there are several 100, 000 images per examination, and a very experienced endoscopist needs between one and several hours to view and analyze all the video data [29]. Due to this, it is important to develop automatic methods that can reduce the burden on physicians and speed up the process of video analysis. We therefore also developed an efficient semi-automatic annotation subsystem [2]. This tool makes it easy for doctors to annotate and provide data to the system. The manual annotations of the doctors are combined with semi-automatic methods that extend the provided data. Our semi-automatic process reduces the time that time physicians spend on annotating. Instead of annotating every frame, they can provide annotations on a single frame of an image series or video. They identify abnormalities, mark a region of interest and tag it accordingly. The automatic step [2] uses this information to track the regions of interest on subsequent as well as previous frames. Due to the fact that the medical doctor is usually located in a hospital with security restrictions, the implementation of the software is done with standard web technologies which do not require any installation at the hospitals systems. This also includes the storing of all information on the systems side and moves the responsibility of maintaining the system and data integrity from the user to the system. Besides getting data for the system to enable automatic screening, the annotation subsystem makes it possible to use the annotated videos in a medical video archive for surgical documentation or teaching purposes.

3.2 Automatic Detection Subsystem

The subsystem for detection and automatic analysis is designed in a modular manner, so that it can be easily extended to additional diseases, to new subcategories of a disease, as well as newly requested information, such as determining a polyp's size. At the moment, the subsystem consists of two parts, the detection subsystem that detects frames containing irregularities, and the localisation subsystem that localises the position of the irregularity within a detected frame.

3.2.1 Detection Subsystem. The detection subsystem detects whether a frame contains an irregularity, without any indication of a position of this irregularity in the frame. The detection of specific abnormality type can be performed after the initial training of the detection subsystem using previously collected training frames set. All frames that are used in training are divided into two disjoint sets. These two sets contain example images for abnormalities and images without any abnormality. Each of these sets can be seen as the model for a specific disease.

The detection subsystem supports a hierarchical concept of models and sub-models. This does allow it to, for example, first detect a polyp and then distinguish between a polyp posing a low or high risk of developing into CRC using the *NICE* classification [22]. To compare and determine the abnormalities in a given frame, we use global image features. In previous work [45], we showed that, in case of only detecting whether a frame contains an irregularity or not, global features can outperform local features, i.e., at least reach the same results with respect to detection and significantly outperform local features in terms of processing speed.

The whole system is built using the Lire [31] open source library for content-based image retrieval, written in *Java*. This library provides a comprehensive set of tested algorithms to extract a variety of global image features. It allows us to experiment with a wide range of global image features for detecting or clustering video frames from colonoscopy or WVC videos. Lire uses *Lucene indexes* [16] for storing and searching image feature data.

Indexing. The index structure is field- and row-based. Each row is defined by its fields, e.g., the image path, the binary values for the feature or the hash representation of the feature, etc. The number of fields and their size are variable depending on the number or type of feature. All feature values are stored as byte representation of the respective feature vector as well as a text field containing hash values from a random projection hashing [31] approach.

The hashing approach is based on locality sensitive hashing [31] (LSH). The main idea is to use multiple random hash functions to hash the values of the features giving the same hash values for the similar images. This is done by a linear projection in random directions of the hash functions in the feature space of the image. The created hash codes are ineffective and a large number of hash tables is needed to achieve a reasonable search quality, but compared to the increased speed of the algorithm these are minor disadvantages that can be ignored [49].

We use a hash function $h(v) \in \{0, 1\}$, which is defined for a histogram v as $h(v) = sgn(v \cdot r)$, where sgn is the sign function (extracts the sign of a real number) and r is a random vector with uniformly distributed elements r_i with $-w \leq r_i \leq w$. n hash functions are combined as a bit string in one single hash value $H(v) < 2^n$. For indexing m hash values, m functions $H_j(v)$, $0 \leq j < m$ are generated.

The parameters for the hashing-based approximate indexing are chosen based on evaluations on an image dataset consisting of 10^5 images. To achieve a good performance for precision and search time, the parameters have been set as following: $w = 2$, $n = 12$, and $m = 150$. This leads to a significant speed-up and at the same time, to a good trade-off between search time and precision.

Search. The search for an image that we use in our search-based algorithm is performed on the fly on the previously created indexes. For each image, a term-based query from the hashed feature values of the query image is created, and a comparison with all images in the index is performed resulting in a ranked list of similar images. The ranked list is sorted by a distance or dissimilarity function associated with the low level features. This is done by computing the distance between the query image and all images in the index. The distance function for our ranking is the *Tanimoto* distance [54], which is computed by taking the ratio of the number of elements that intersect and the union of the elements:

$$f(A, B) : [0, 1]^n x [0, 1]^n \rightarrow \mathbb{N} = \frac{A \cdot B}{|A|^2 + |B|^2 - A \cdot B}$$

A smaller distance between an image in the index and the query image means a better rank [54]. The final ranked list is used in the

classification step. To be able to classify an image efficiently, two important aspects have to be considered: the selected features and the feature combination.

Feature Selection. Different features have different properties, and they are therefore useful in different scenarios. To make the search-based classifier fast and accurate, we have to decide which features we want to use for a specific use case, because a random selection of global features and random combinations of feature can lead to negative results for the classification or search task. Badly chosen feature combinations can introduce noise (if too many features are combined and some of them do not add any information to the classification problem) or make the search slow (if the index is very big because of too many used global features). A lot of work has been performed in the field of feature selection, and different machine learning techniques were utilized for it [35]. For example, an information gain (IG) attribute evaluation, which computes the information gain of a given feature with respect to the classification problem to determine which feature gives the most information [12]. Another example is the SVM attribute evaluation, which ranks the variables of the features using a weight assigned from a support vector machine [19]. Furthermore, Guldogan and Gabbouj [18] tried to utilize standard feature selection algorithms, like IG, to measure a features performance for a given task. Based on these measurements they applied majority voting to produce a ranked list of features further used to select the best working ones. Their evaluation results demonstrate that this method can improve the classification performance and at the same time reduce the computation time.

Currently, we perform a simple feature selection by testing different combinations of features on smaller reference datasets to find the best combinations in terms of processing speed and classification accuracy. For the further system improvement, we will implement several advanced features selection algorithm and will perform a comparison in order to select the best for our use case.

Feature Combination. Features can be combined in two different ways. The first is called feature values fusion or *early fusion*, and it basically fuses values of different features into a single representation before they are used in a decision-making step. The second one is called decision fusion or *late fusion* where the features are combined after a decision-making step. Our system implements feature combination using the *late fusion* approach.

Search-based Classification. The search-based algorithm developed in this work has been implemented using *Lire*. Since *Lire* is based on the *Lucene indexes* [16], it also allowed us to create an algorithm that is able to include any type of multimedia data if needed. *Lucene inverted indexes* are created using k-way merge [16]. The index segments are sorted in memory and then merged. Each newly added data element is treated as a new segment and added to existing segments. These indexes have the advantage that they are fast to update and reasonably fast to search. The indexes are field-based and the number of fields is variable depending on the number of used features. The fields are stored using LSH as described before. The algorithm is basically a simple K-NN algorithm, which defines classes c as:

$$c = \arg\max_{\hat{c} \in C} \{ClassScore(\hat{c})\}$$

$ClassScore$ is calculated by summing up the occurrences of each class c and multiplying it with the summed $WeightedRankScore$. $RankScore$ per class is calculated by dividing 1 by the rank for each search query.

$$ClassScore(c) = |c| \sum_{I_i \in \{I_i | Class(I_i) = c\}} RankScore(I_i)^{-1}$$

The $WeightedRankScore$ is the sum of all $RankScore$s in the rank list. This algorithm can be used for supervised and unsupervised learning, two or multi-class classification and different types of input data ranging from features extracted from images to videos to meta data. Its main advantages are its simplicity, that it achieves state-of-the-art classification results and that it is very fast in terms of processing time. The latter is demonstrated by applying it to different use cases described in the following section.

Implementation Details. The indexer is created as a separate tool and in a way that it is easy to distribute over different nodes using, for example, Apache Storm. Indexing is performed when the training data is inserted into the system and is suited for batch processing. Creating the models for the classifier can be done offline and does not influence the real-time capability of the system because it is only done once at the very first time when the training data is inserted into the system. It creates indexes for all directories passed on from the system. The visual features to calculate and store in the indexes can be chosen based on the abnormality because, for different types of diseases, different set of features or combinations are better. For example, bleeding is easier to detect using color features, whereas polyps require also shape and texture information. The indexer stores the generated indexes in a subdirectory inside the indexed directory. If multiple directories are passed for indexing, it creates a separate index for each directory.

The classifier can be used to classify video frames from an input video into as many classes as the detection subsystem model consists of. The classifier uses indexes generated by the indexer as described before. In contrast to other classifiers that are commonly used, this classifier is not trained in a separate learning step. Instead, the classifier searches previously generated indexes, which can be seen as the model, for similar visual features. The output is weighted based on the ranked list of the search results. Based on this, a decision is made. We refer to these previously generated indexes, which are searched for similar image features, as classifier indexes or indexes containing training data. The classifier expects at least one classifier index and an input source. The input source can either be a video, an image or another previously generated index. The classifier also includes a benchmarking function that will output the evaluation information and an HTML page with a visual representation of the results, once the processing is finished. The classifier is parallelized and allows to choose how many CPU cores are used to process the data. In the future, a GPU implementation will be supported, because our previous research [44, 46] showed that it can significantly improve the performance.

We have released the source code of the detection subsystem as an open-source project called *OpenSea*[1], under the terms of the GPL version 3[2].

[1] https://bitbucket.org/mpg_projects/opensea
[2] http://www.gnu.org/licenses/gpl-3.0.en.html

3.2.2 Multi-disease Classification. Previously, we claimed that one major difference between our system and related approaches is that it can easily be extended to detect other endoscopic findings (abnormalities, diseases, anatomic landmark or other relevant events during the examination of a patient). To prove that our system is able to perform multi-class classification for diseases beyond polyps, we developed a detection prototype that implements two approaches: global-feature-based and deep-learning-based. Both approaches are tested on a dataset collected from the Bærum Hospital in Norway, one of our collaborators. The amount of data that has been annotated to evaluate the multi-class classification is rather limited so far, and consequently, these results are preliminary.

Multi-class global-feature-based approach (GF-classifier). The basic search-based classification part of the system is used to create a separate classifier for each disease that we want to classify. The difference to the initial version of the detection part is that the ranked lists of each search-based classifier are used in an additional added classification step to determine the final class. For the final classification, we use the random forest classifier (RFC) [7]. It is important to point out that other classification algorithm could be used, and that we choose the random forest approach because it is fast while achieving good results [56].

The RFC creates a forest of classification trees. Each tree is a decision tree that makes, at each of its inner nodes, a branching decision based on one or more feature dimensions. The conditions for these branching decisions are randomly created at the time of the tree's creation, and applied deterministically afterwards. Thus, classes are randomly defined, but features are deterministically classified. To determine the final class, the classifier combines all decisions trees into a final decision using the same late fusion technique used for the features in the standard search-based classifier.

RFC allows parallel classification for each of the separate random trees of the forest. Apart from that the parallel step does also allow for very fast training. Further, the RFC is very efficient for large datasets because of the ability to find distinctive classes in the dataset and also to detect the correlation between these classes. The disadvantage is that training time increases linearly with the number of trees. However, this is not a problem for our use case since training time is not critical. We use the RFC implementation provided by the Weka machine learning library [20].

Multi-class deep-learning-based approach (Deep-classifier). The deep-learning-based classification approach is implemented using Google Tensorflow [1]. As a basis for the deep learning network architecture, we use Inception v3 [52], which is a modern neural network designed for image classification tasks. The Inception v3 model is pre-trained on the ImageNet dataset [13]. From the Inception v3 model, we removed the last layer and retrained it with our medical image classes following the approach presented in [14]. This makes it possible to reuse visual concepts learned from the ImageNet dataset to perform the learning on a smaller dataset.

After removing the final layer from the model, we insert a randomly initialized fully connected layer and retrain the final layer from scratch. All the other layers do not change. This comes with the advantages that not so much training data is needed to train the network, which is a benefit for our medical scenario where lack of good data is a common problem, and that it is faster. Its takes around one day with our settings to retrain the model. The re-trainer is based on an open source implementation [1] of Tensorflow.

At first, we calculate for each image the values for the second last layer (also called bottleneck), which can be seen as kind of features representing the images. These features are then used to retrain the final layer of the network based on the new classes using a softmax function [5]. For the retraining, we run $10,000$ training steps. Each step takes 20 random images in their pre-extracted feature representation to retrain the layer. Because of the small amount of training data, we also perform distortion operations on the images, which is required to avoid network overfitting. In more detail, we perform random cropping, random rescaling and random change of brightness. The grade of distortion is set to 25% per image. In the case of polyp detection, distortions will not destroy the meaning of the image (like it would do if someone, for example, wants to detect letters). After the model has been retrained, it is used as a multi-class classifier that provides the top five classes based on probability for each class.

3.2.3 Localisation Subsystem. The localisation subsystem is intended for finding the exact position of irregularities, which is used to show markers on the disease in the visualization subsystem. All images that we process during the localisation step come from the positive frames list generated by the detection subsystem. The processing of the images is implemented as a sequence of intraframe pre- and main-filters.

Pre-filtering is needed because we use local image features to find the exact position of objects in the frames. Irregularities can have different shapes, textures, colors and orientations. They can be located anywhere in the frame and also partially be hidden and covered by biological substances, like for example seeds or stool, and lighted by direct and reflected light. Moreover, the image itself can be interleaved, noisy, blurry and over- or under-exposed, and it can contain borders, subimages and a lot of specular reflections (flares) caused by endoscope's light source. Images can have also various resolutions depending on the type of endoscopy equipment used. All these nuances negatively affect the local features detection methods and have to be specially treated to reduce localisation precision impact. In our case, sequence of filters are used to prepare raw input images for the following analysis. These processing steps are border and subimage removal, flare masking and low-pass filtering. After pre-filtering, the images are used for the following local features analysis.

At the moment, we have only implemented localisation of colon polyps using our local feature approach. For future work, we aiming to also localize other irregularities like cancer, bleeding, parasites, etc. The main idea of the localisation algorithm is to use the polyps' physical shape to find the exact position in the frame. In most cases, the polyps have the shape of a hill located on a relatively flat underlying surface, or the shape of a round rock connected to an underlying surface with legs varying in thickness. These polyps can be approximated by an elliptically shaped region that consists of local features that differ from the surrounding tissue.

To detect polyps, we use the following sequence of filters: binary noise reduction filter, 2D-gradient filter, threshold border detection filter and binary noise removal filter. The next step creates a filtered binary contour image approximated by a set of ellipses. The

precision of contours approximation via ellipses is measured as distance from ellipses' borders to contours' pixels, which results in an energy map. The final coordinates of one or more polyps in the frame are chosen by looking for maxima in the energy map. For performance reasons, the localiser is implemented in C/C++ and uses *OpenCV* [6]. An example of the output is shown in figure 2.

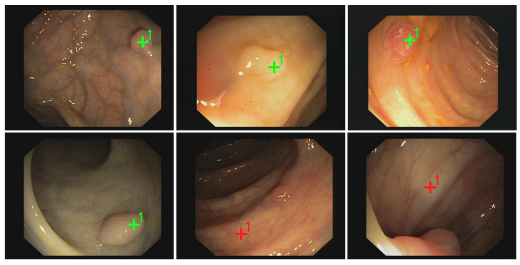


Figure 2: Output of the localisation subsystem marking the possible locations of polyps. The first 4 frames show an exact match, the last two show false positives.

3.3 Visualization and Computer Aided Diagnosis Subsystem

This subsystem has two main purposes. Firstly, it should help in evaluating the performance of the system and get better insights into reason for successes and failures. Secondly, it can be used as a computer-aided diagnosis system for medical experts.

First, we have the *TagAndTrack* subsystem [2] that can be used as a visualisation and computer-aided diagnosis system. Second, we developed an open-source application *ClusterTag* [43] designed for interactive exploration and labeling of big image collections in conjunction with semi-automatic image clustering, annotation and tagging. Third, we developed a web-based visualization that can also be used to support medical experts and is easy to use and distribute. It takes the output of the detection and localisation subsystems and creates a web based visualisation, which later may be combined with a video sharing platform [21, 51], where doctors are able to watch, archive, annotate and share information.

4 SYSTEM EVALUATION

We tested the whole system in terms of accuracy and system performance. For all measurements, we used the same computer (32 cores AMD Opteron 8218 Linux server, 128GB RAM, from 2006). For all experiments, we used the ASU Mayo Clinic polyp database[3]. This is currently the biggest publicly available dataset consisting of 20 videos (converted from WMV to MPEG-4 for the experiments) with a total of 18, 781 frames and different resolution up to full HD (1920x1080) [53].

4.1 Detection and Localisation Accuracy

For detection and localisation accuracy, we used the common metrics, precision, recall and F1 score. All experiments have been conducted on the complete ASU Mayo Clinic polyp database and each subsystem has been evaluated separately.

[3]https://polyp.grand-challenge.org/site/Polyp/AsuMayo/

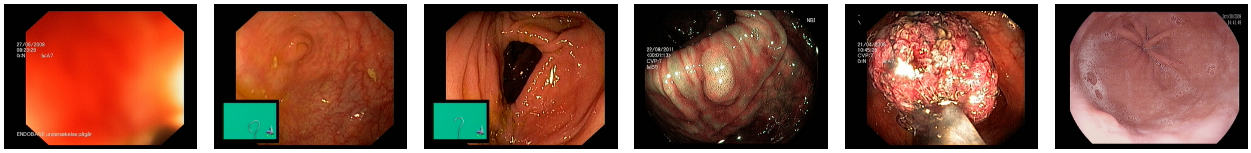
4.1.1 Detection Accuracy. We conducted a leave-one-out cross-validation to evaluate the detection subsystem. This is a method that assesses the generalization of a predictive model. In our case, it describes the process where the training and testing datasets are rotated, leaving out a single different non-overlapping item or portion for testing, and using the remaining items for training. This process is repeated until every item or portion has been used for testing exactly once [15]. Our system allows us to use several different global image features for the classification. The more image features we use, the more computationally expensive the classification becomes. Further, not all image features are equally important or provide equally good results for our purpose. As a first step, we therefore needed to find out which image features we want to use for classification, and we ran the detection with all possible image features in Lire [32] selected on a dataset. Based on this evaluation, feature extractors and descriptors according to Joint Composite Descriptor (JCD) [32] and Tamura [32] (in the following simply called *features* for brevity) were chosen for our measurements due to their promising performance.

To assess the actual performance of the classifier using these two features, we conducted a leave-one-out cross-validation with all available video sequences. With these settings, we achieved an average precision of 0.889, an average recall of 0.964 and an average F1 score value of 0.916. The problem with this average calculation is that different video sequences contribute values based on different numbers of video frames. If we weight the values contributed by every single video sequence with the amount of frames in the sequence, we achieve an average precision of 0.9388, an average recall of 0.9850, and an average F1 score value of 0.9613. In other words, the results mean that we can detect polyps with a precision of almost 94% and we detect almost 99% of all polyp-containing frames. The evaluation is presented in table 1.

Table 1: Performance evaluation by leave-one-out cross-validation for all available videos, using JCD and Tamura features.

Video	True positive	True negative	False positive	False negative	Precision	Recall	F1 score
np_5	1	680	0	0	1	1	1
np_6	1	836	0	0	1	1	1
np_7	1	767	0	0	1	1	1
np_8	1	710	0	0	1	1	1
np_9	1	1,841	0	0	1	1	1
np_10	1	1,923	0	0	1	1	1
np_11	1	1,548	0	0	1	1	1
np_12	1	1,738	0	0	1	1	1
np_13	1	1,800	0	0	1	1	1
np_14	1	1,637	0	0	1	1	1
wp_2	140	9	20	70	0.875	0.6666	0.7567
wp_4	908	1	0	0	1	1	1
wp_24	310	68	127	12	0.7093	0.9627	0.8168
wp_49	421	12	62	4	0.8716	0.9905	0.9273
wp_52	688	101	284	31	0.7078	0.9568	0.8137
wp_61	162	10	165	0	0.4954	1	0.6625
wp_66	223	12	165	16	0.5747	0.9330	0.7113
wp_68	172	51	20	14	0.8958	0.9247	0.9100
wp_69	265	185	138	26	0.6575	0.9106	0.7636
wp_70	379	1	0	29	1	0.9289	0.9631
Weighted average:					0.9388	0.9850	0.9613

4.1.2 Multi-class Classification Accuracy. To evaluate the multi-class classifiers, we collected a new dataset from one of our partner hospitals. The dataset contains six different endoscopic findings that can occur during a colonoscopy with 50 images each, which leads to



(a) Blurry frame (b) Cecum (c) Normal colon mucosa (d) Polyp (e) Tumor (f) Z-line

Figure 3: Example for anatomic findings (classes) in the multi-class dataset.

a total number of 300 images[4]. The classes in the dataset are blurry frames, cecum (pouch that is the beginning of the large intestine), normal colon mucosa (healthy colon wall), polyp, tumor, and Z-line (an anatomic landmark in the colon than can help doctors to orientate). Figure 3 shows one example for each class in the dataset. Because of the small number of images in the dataset, we performed cross-validation. For the cross-validation, we randomly separated the images into 10 different sets of training and test data. Each training and test subset contains 25 images per class. Multi-class classification is then performed on all 10 splits and then combined and averaged. Following this strategy even with a smaller number of images, a quite accurate estimation about the performance can be made.

Table 2 shows the confusion matrix (a standard tool for evaluating multi-class classifiers showing the actual class compared to the detected class) for the GF-classifier. The results are a clear indication that this approach performs well. An interesting insight is that normal colon mucosa is often miss-classified as cecum (cecum is also sometimes miss-classified as normal colon mucosa). The example images for cecum (figure 3(b)) and normal colon mucosa (figure 3(c)) reveal that this is not very surprising since it is even hard for a human observer to make a clear decision. Furthermore, from a medical point of view, normal colon mucosa are part of the cecum and under real-world circumstances, this would not be a relevant mistake.

Table 2: Confusion matrix and standard metrics for the six-class classification performance for the multi-class global-features-based approach. The classes are Blurry frames (A), Cecum (B), Normal colon mucosa (C), Polyps (D), Tumor (E), Z-line (F).

		Detected class						*Metrics*		
		A	B	C	D	E	F	Precision	Recall Sensitivity	F1-score
Actual class	A	**250**	0	0	0	0	0	1.0	1.0	1.0
	B	0	**226**	21	3	0	0	0.704	0.904	0.791
	C	0	85	**165**	0	0	0	0.85	0.66	0.743
	D	0	10	8	**226**	6	0	0.953	0.904	0.928
	E	0	0	0	8	**242**	0	0.975	0.968	0.971
	F	0	0	0	0	0	**250**	1.0	1.0	1.0
							Average	**0.914**	**0.906**	**0.91**

The performance of Deep-classifier, which is presented in table 3 can also be considered as good. This approach confuses the classes polyp and cecum more than the GF-classifier, but it is better in detecting normal colon mucosa. For detecting blurry frames and Z-lines, it performs at the same level as the GF-classifier. Based on the confusion matrix for both approaches, we can see that for some classes, the GF-classifier is better and for other classes the Deep-classifier.

[4]The dataset that we could collect in the given time frame with the help of our medical partners is rather small, but it is large enough for a proof-of-concept in combination with cross validation.

Table 3: Confusion matrix and standard metrics for the six-classes detection performance evaluation for the deep-learning-based approach.

		Detected class						*Metrics*		
		A	B	C	D	E	F	Precision	Recall Sensitivity	F1-score
Actual class	A	**250**	0	0	0	0	0	1.0	1.0	1.0
	B	0	**183**	64	3	0	0	0.782	0.732	0.756
	C	0	34	**197**	19	0	0	0.641	0.788	0.707
	D	1	17	45	**183**	4	0	0.875	0.732	0.797
	E	0	0	1	4	**245**	0	0.983	0.98	0.981
	F	0	0	0	0	0	**250**	1.0	1.0	1.0
							Average	**0.879**	**0.872**	**0.876**

Comparison of the GF- and the Deep-classifiers using the standard metrics including precision, recall/sensitivity and F1-score reveals that the GF-classifier outperforms Deep-classifier significantly with a precision of 0, 914, a recall of 0, 906 and a F1-score of 0.91 for the GF-classifier compared to a precision of 0, 879, a recall of 0, 872 and a F1-score of 0.876 for the Deep-classifier.

4.1.3 Localisation Accuracy. Table 4 shows the performance of the localisation subsystem. As ground truth, we used the exact positions of the polyps as provided in the ASU Mayo clinic polyp database. Overall, we reached an average precision of 0.3207, a recall of 0.3183 and an F1 score of 0.3195. The values seem to be rather low, but it is important to point out, that the current localisation algorithm outputs four possible locations per frame. Currently, we are working on an implementation that will be able to output only one location per frame.

Table 4: Performance evaluation of the localisation algorithm in terms of accuracy.

Dataset	True positive	False positive	False negative	Precision	Recall	F1 score
CVC-ClinicDB	397	215	249	0.6487	0.6146	0.6312
ASUMayo 2	1	244	244	0.0041	0.0041	0.0041
ASUMayo 4	443	467	467	0.4868	0.4868	0.4868
ASUMayo 24	74	300	300	0.1979	0.1979	0.1979
ASUMayo 49	36	355	355	0.0921	0.0921	0.0921
ASUMayo 52	194	490	490	0.2836	0.2836	0.2836
ASUMayo 61	129	80	80	0.6172	0.6172	0.6172
ASUMayo 66	92	142	142	0.3932	0.3932	0.3932
ASUMayo 68	63	126	126	0.3333	0.3333	0.3333
ASUMayo 69	0	235	235	0.0000	0.0000	0.0000
ASUMayo 70	4	381	381	0.0104	0.0104	0.0104
Average:				0.3207	0.3183	0.3195

4.2 System Performance

One further requirement for the system is performance. The idea is, as mentioned before, to use the system during live colonoscopies and for mass screening for irregularities in the GI tract, using video sequences, recorded by colonoscopes or WVCs.

For the evaluation, we decided to use the configuration of the system that performed best in the accuracy experiment, because

this scenario will be used in the live system setup, i.e., the global-feature-based version. To enable live assistance for endoscopies, we must reach a frame rate of at least 25 frames per second. For all tests, we used three videos from three different endoscopic devices and different resolutions. The three videos are wp_4 with $1,920x1,080$ and 910 frames, wp_52 with $856x480$ and $1,106$ frames and np_9 with $712x480$ and $1,843$. We chose these three videos because they provide representative examples of the video resolution variations for different types of endoscopic devices.

4.2.1 CPU Processing. For the detection approach, we first measured the indexing part that creates the model that is later on used by the classifier. This process has no real-time requirement and can be seen as batch processing, but it should be feasible for larger datasets. Extracting two features and indexing them for the whole ASU Mayo dataset takes on average 8 milliseconds per frame. There is no big difference between the indexing time of different resolutions. We tested the scaling potential by indexing different datasets. The first dataset $D1$ contains $3,871$ frames, the second one $D2$ contains $14,909$ frames, the third one $D3$ contains $29,818$ frames and the last one $D4$ with $100,000$ frames. Table 5 shows the overall results. We found that a larger dataset leads to a faster indexing time per frame, that is caused by runtime Java code optimizer. Furthermore, we did not find a processing speed increase after more than $30,000$ frames in the dataset. Further processing speed increase is limited by the I/O bottleneck since increasing the number of cores did not increase performance. All in all, our experiments show that the indexer is scalable, can be used with big datasets and it should meet all requirements of the system for future tasks.

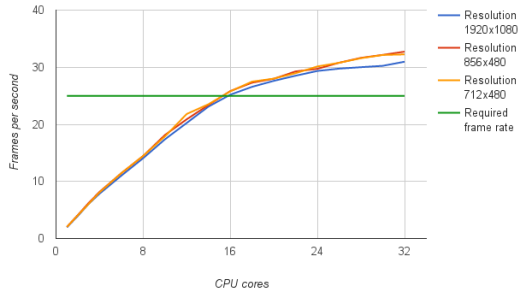
Table 5: Performance evaluation of the indexing part. 4 different datasets with different sizes have been tested to show the scaling capability of the indexing part.

Index	frames	total time in seconds	time per frame in ms
$D1$	$3,871$	89.78	23.1
$D2$	$14,909$	178.55	11.9
$D3$	$29,818$	231.75	7.7
$D4$	$100,000$	782.351	7.8

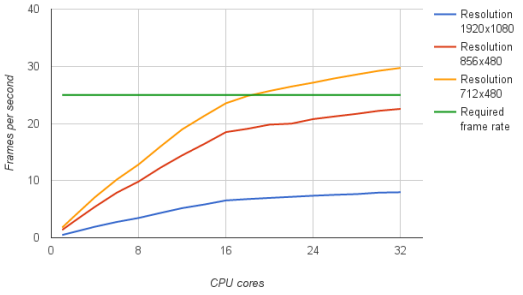
The performance of the detection is more important, since the system should process frames at 25 fps or better to make it usable for live applications. For all tests, we used the 3 different videos described before. Figure 4(a) shows the detection subsystem's performance for the tested videos. The required frames per second for all three resolutions are reached with 16 CPU cores.

Figure 4(b) shows the localisation subsystem's performance for all videos. The required frame rate is not reached for the highest resolution and the best result is 7.9 frames per second. The same is true for the resolution of 856×480. The required frames per second for the lowest resolution are reached with 19 CPU cores used in parallel. The outcome of these experiments clearly shows that our system also can reach real-time requirements for the localisation subsystem but that we need to improve the performance for higher resolutions.

4.2.2 Memory. Figure 5(a) and figure 5(b) show the memory usage for both subsystems. In the localisation, the memory usage behaves normally and shows that the localisation is scalable in terms of memory. For the detection subsystem, the memory usage



(a) The detection subsystem FPS.



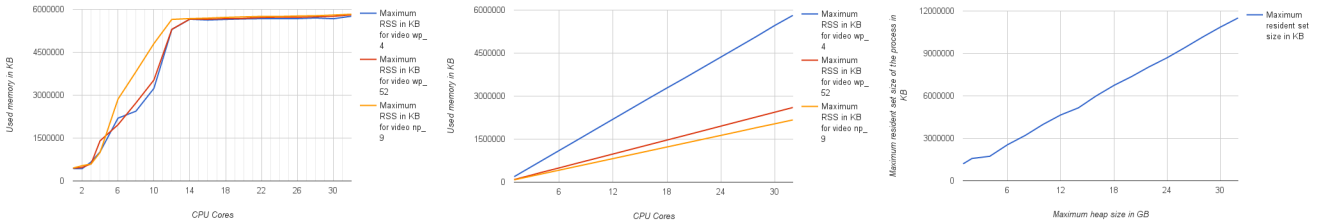
(b) The localisation subsystem FPS.

Figure 4: System performance in terms of frames per second (FPS) depending on the number of CPU cores and the resolution of the videos.

shows an interesting behavior after a certain number of used CPU cores. Therefore, a closer look into it was necessary.

Figure 5(c) depicts this closer look into the detection subsystem memory performance. We tested different memory sizes used for the detection starting from 1GB up to 32GB. This shows that the available memory for the detection part does not influence the frames per second performance. The Java memory scheduler uses as much memory as it can get, but it also performs well with only 1GB. This proves that the detection part does not depend on memory, and therefore, memory is not a bottleneck for scaling.

4.2.3 Size of the Index. A final question that we wanted to answer is if the size of the used classification indexes (number of indexed examples) influences the detection accuracy or system performance. Figure 6 shows the system performance in terms of detection accuracy (F1 score) and frames per second for 3 different training data sizes. The expectation was that smaller indexes would lead to a higher frames per second throughput but with a loss of classification performance. The experiment showed that the index size did not have a significant influence on the number of frames per second output of the detection system. It is possible that an index with several hundred thousand of frames will most probably lead to a lower frames per second output. But, in the intended medical field, a lack of training data is normal. Therefore, this will not influence our system. Another positive aspect is that the classification performance does not decrease with smaller indexes. It is even the opposite, because for wp_52, the F1 score increased slightly compared to the full training data. This shows that the



(a) Memory consumption in the detection part. (b) Memory consumption in the localisation part. (c) Memory consumption versus heap size.

Figure 5: System benchmarks of memory usage.

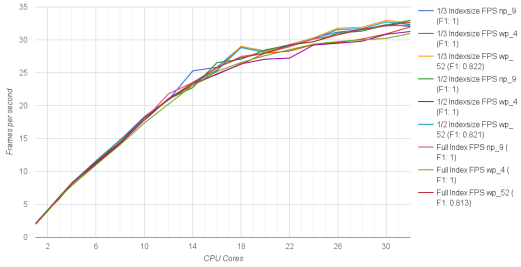


Figure 6: This chart shows how the amount of training data influences the performance of the detection subsystem in terms of frames per second output. The training data has been reduced to 1/2 of the original size (ca. 8, 800 **frames) and 1/3 (ca.** 5, 800 **frames).**

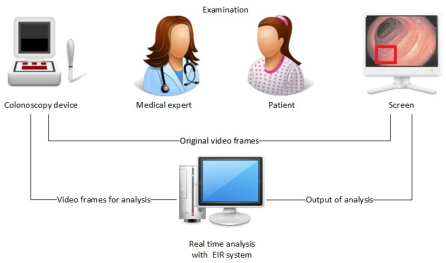


Figure 7: The planned structure of the live system. The medical expert doing a normal examination is assisted in real-time with the results of the video analysis displayed on the auxiliary screen.

detection subsystem also performs very well with a smaller amount of training data matching well our medical scenario.

4.2.4 Initial Cloud Experiments. To investigate what the performance would be on actual hardware for the detection subsystem, some initial tests on Amazon AWS EC2 instances were conducted. On a *c4.8xlarge* instance (Intel Xeon E5-2666-V2 with 36 virtual CPU cores), we were able to classify a video (MPEG-4) with 1, 924 frames and a resolution of $1,920 \times 1,080$ with the features JCD and Tamura, in 29.377 seconds with 65.5 fps. When classifying data from a raw video file the processing time increased to 39.599 seconds with 48.6 fps. When reading the data from a Windows media video (wmv) file, the processing time increased to 40.452 seconds with 47.6 fps. The *c4.8xlarge* instance is the most powerful instance offered by Amazon. We therefore conducted the same tests also on a less powerful *c4.4xlarge* instance (Intel Xeon E5-2666-V2 with 16 virtual CPU cores). Using this instance, we were able to process the MPEG-4 video data in 60.19 seconds with 31.97 fps, the wmv file in 81.17 seconds with 23.7 fps and the raw video file in 79.718 seconds with 24.14 fps. This shows that on newer hardware an even better performance can be achieved.

5 REAL WORLD USE CASES

In this section, we will describe two real world use cases where the presented system can be used. The first one is a live system that will support medical doctors during endoscopies. Currently, we are working on setting it up in one of our partner hospitals. The second one is a system that will automatically analyse videos captured by WVCs. Several hospitals all over Europe and US are involved in this part, and currently, we are collecting data. The first use case requires fast and reliable processing, and the second requires a system that is able to process a large amount of data in a reliable and scalable way.

5.1 Live System

Figure 7 gives an overview of the proposed live system. Live endoscopy is a common GI examination and is essential for the diagnosis of most mucosal diseases in the gastrointestinal tract, particularly diagnosis of CRC and its precursors. The aim of the live system is to put it between the screen of the doctor and the endoscopy processor. While the endoscopist performs the colonoscopy, the system analyses the video frames that are recorded by the colonoscope. First, we planed to optically show the physician (for example with a red or green frame around the video) when the system detects something abnormal in the actual frame. This can also be extended to determine which disease that the system most probably detected and provide this information to the doctor. Apart from supporting the medical expert during the colonoscopy, the system can also be used to document the procedure. After the colonoscopy, an overview can be given to the doctors where they can make changes or corrections, and add additional information. This can then be stored for later purposes or used in a written endoscopy report. Further, it would be practical to store high quality images of the most important parts. As paper [11] shows, single images can be an efficient way to store important findings from an examination.

5.2 Wireless Video Capsule Endoscope

The present WVCs have a resolution of 256*x*256 with 3-10 frames per second (adaptive frame rate with a feedback loop from the receiver to the transmitter). They do not have optimum lighting,

making it difficult use the images. Nevertheless, ongoing work tries to improve the state-of-the-art technology, which will make it possible to use the methods and algorithms developed for colonoscopies also for WVCs [8, 27].

The multi-sensor WVC is swallowed in order to visualize the GI tract for subsequent diagnosis and detection of GI diseases. Thus, people will be able to buy WVCs at the pharmacy, and connect and deliver the video stream from the GI tract to the phone over a wireless network. The video footage can be processed in the phone or delivered to our system, which finally analyses the video automatically. In the best case, the first screening results are available within eight hours after swallowing the WVC, which is the time the camera typically spends traversing the GI tract.

In order to develop such a system, many unsolved tasks need to be addressed through (interdisciplinary) research and development. For example, the training and learning step that allows the system to detect different disease in the GI tract. In the case of the colon, accuracy of existing methods is far below the required precision and recall, and the processing of the algorithms does not scale in terms of big data. Each type of disease or irregularity requires interaction between medical researchers dictating what the system must learn to detect, image processing researchers investigating detection or summarization algorithms, hardware developers to develop/produce/research sensors, distributed processing researchers in order to scale and distribute the (big data) analytics and processing of the sensor data. For other scenarios, like in the upper part of the GI tract, there will be similar challenges and corresponding interaction between research disciplines.

Obviously, the project has high and ambitious goals in developing an end-to-end solution where data recorded by next generation camera and WVCs automatically are processed and algorithmically analyzed for potential pathology in the GI tract. There are large challenges with respect to accuracy (precision and recall), scale of the processing and hardware data quality because of different manufacturers (Olympus and Given are the biggest ones). The aim is to be a leading contributor in the area of medical imaging and sensor processing in the GI tract as well as storing, processing and analysing this type of data. Such next-generation big data applications in the area of medicine are frontiers for innovation and productivity in health systems where there are currently large initiatives both in the EU and the US.

6 CONCLUSION

In this paper, a multimedia system for disease detection and classification in the GI tract has been presented. We briefly described the whole pipeline of the system from annotation (data collection for system learning) to visualisation (doctor feedback). We introduced two new multi-class classification methods, based on global image features and deep learning neural networks. The novelty of the research includes the implementation of a whole system pipeline as a combination of many existing components, as well as several new ones. A detailed evaluation in terms of detection and localisation accuracy and system performance has been performed, and we meet the requirements listed in section 2: (i) high detection accuracy with an F1 score of 96% for polyps, (ii) real-time processing to support live examinations like colonoscopies with a frame rate between 30-65 on the given hardware, (iii) efficient resource utilization to allow massive scale using WVCs shown by both the real-time processing and the low memory consumption, and (iv) expandability to allow the system to support new diseases as shown by the high accuracy multi-disease detection experiment. Our experiments show that the proposed system can achieve equal results to state-of-the-art methods in terms of detection accuracy. Further, we showed that the system outperforms state-of-the-art systems in terms of system performance, that it scales in terms of data throughput and that it can be used in a real-time scenario. We also presented automatic analysis of WVC videos and live support of colonoscopies as two real world use cases that will benefit from the proposed system and will actually be tested and used in our partner hospitals.

For future work, we plan to improve the detection and localisation accuracy of the system, including even more different abnormalities to detect and work on the localization of irregularities beyond polyps. Presently, we are working with medical experts to collect more training data. As a first result, we just finished two new datasets: an extended multi-class image-dataset for computer aided GI disease detection called Kvasir [42] and a new bowel (colon) preparation quality video dataset called Nerthus [41]. Both datasets are released under open-source and can be used by the community. Additionally, we work on the set-up of the real world use case in the hospitals. Finally, to further improve the performance of the system, we work on an extension that allows the system to use GPUs to further utilize the parallelization potential of the workload.

REFERENCES

[1] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and others. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org* 1 (2015).

[2] Zeno Albisser, Michael Riegler, Pål Halvorsen, Jiang Zhou, Carsten Griwodz, Ilangko Balasingham, and Cathal Gurrin. 2015. Expert Driven Semi-supervised Elucidation Tool for Medical Endoscopic Videos. In *Proc. of MMSys*. 73–76.

[3] Luís A Alexandre, Joao Casteleiro, and Nuno Nobreinst. 2007. Polyp detection in endoscopic video using SVMs. In *Proc. of PKDD*. 358–365.

[4] Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. 2009. Texture-based polyp detection in colonoscopy. In *Proc. of BfM*. 346–350.

[5] Christopher M Bishop. 2006. Pattern recognition. *Machine Learning* 128 (2006).

[6] Gary Bradski and Adrian Kaehler. 2008. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Incorporated.

[7] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[8] Rohit Chandra and Ilangko Balasingham. 2015. A microwave imaging-based 3D localization algorithm for an in-body RF source as in wireless capsule endoscopes. In *Proc. of EMBC*. 4093–4096.

[9] Da-Chuan Cheng, Wen-Chien Ting, Yung-Fu Chen, Qin Pu, and Xiaoyi Jiang. 2008. Colorectal polyps detection using texture features and support vector machine. In *Proc. of MDAISM*. 62–72.

[10] Christine Chin and David E Brown. 2000. Learning in science: A comparison of deep and surface approaches. *Journal of Research in Science Teaching* 37, 2 (2000), 109–138.

[11] Thomas de Lange, Stig Larsen, and Lars Aabakken. 2005. Image documentation of endoscopic findings in ulcerative colitis: photographs or video clips? *GE* 61, 6 (2005), 715–720.

[12] R López De Mántaras. 1991. A distance-based attribute selection measure for decision tree induction. *Machine learning* 6, 1 (1991), 81–92.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*. 248–255.

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.. In *Proc. of ICML*. 647–655.

[15] Bradley Efron and Robert Tibshirani. 1997. Improvements on Cross-Validation: The .632+ Bootstrap Method. *J. Amer. Statist. Assoc.* 92, 438 (1997), pp. 548–560.

[16] The Apache Software Foundation. 2013. Apache Lucene - Index File Formats. (2013). https://lucene.apache.org/core/3_0_3/fileformats.html#Definitions Accessed: 2015-07-29.

[17] B. Giritharan, Xiaohui Yuan, Jianguo Liu, B. Buckles, JungHwan Oh, and Shou Jiang Tang. 2008. Bleeding detection from capsule endoscopy videos. In *Proc. of EMBS*. 4780–4783.

[18] Esin Guldogan and Moncef Gabbouj. 2008. Feature selection for content-based image retrieval. *Signal, Image and Video Processing* 2, 3 (2008), 241–250.

[19] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine learning* 46, 1-3 (2002), 389–422.

[20] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.

[21] Pål Halvorsen, Simen Sægrov, Asgeir Mortensen, David KC Kristensen, Alexander Eichhorn, Magnus Stenhaug, Stian Dahl, Håkon Kvale Stensland, Vamsidhar Reddy Gaddam, Carsten Griwodz, and Dag Johansen. 2013. Bagadus: an integrated system for arena sports analytics: a soccer case study. In *Proc. of MMSYS*. 48–59.

[22] Nana Hayashi, Shinji Tanaka, David G Hewett, Tonya R Kaltenbach, Yasushi Sano, Thierry Ponchon, Brian P Saunders, Douglas K Rex, and Roy M Soetikno. 2013. Endoscopic prediction of deep submucosal invasive carcinoma: validation of the narrow-band imaging international colorectal endoscopic (NICE) classification. *Gastrointestinal endoscopy* 78, 4 (2013), 625–632.

[23] Øyvind Holme, Michael Bretthauer, Atle Fretheim, Jan Odgaard-Jensen, and Geir Hoff. 2013. Flexible sigmoidoscopy versus faecal occult blood testing for colorectal cancer screening in asymptomatic individuals. *The Cochrane Library* (2013).

[24] Sae Hwang, JungHwan Oh, W. Tavanapong, J. Wong, and P.C. de Groen. 2007. Polyp Detection in Colonoscopy Video using Elliptical Shape Feature. In *Proc. of ICIP*. 465–468.

[25] Michal F Kaminski, Jaroslaw Regula, Ewa Kraszewska, Marcin Polkowski, Urszula Wojciechowska, Joanna Didkowska, Maria Zwierko, Maciej Rupinski, Marek P Nowacki, and Eugeniusz Butruk. 2010. Quality indicators for colonoscopy and the risk of interval cancer. *New England Journal of Medicine* 362, 19 (2010), 1795–1803.

[26] J Kang and R Doraiswami. 2003. Real-time image processing system for endoscopic applications. In *Proc. of CCECE*, Vol. 3. 1469–1472.

[27] A Khaleghi and I Balasingham. 2015. Wireless communication link for capsule endoscope at 600 MHz. In *Proc. of EMBC*. 4081–4084.

[28] Baopu Li and M.Q.-H. Meng. 2012. Tumor Recognition in Wireless Capsule Endoscopy Images Using Textural Features and SVM-Based Feature Selection. *IEEE Transactions on Information Technology in Biomedicine* 16, 3 (May 2012), 323–329.

[29] Baopu Li and Max Q. H. Meng. 2009. Computer-based Detection of Bleeding and Ulcer in Wireless Capsule Endoscopy Images by Chromaticity Moments. *CBM* 39, 2 (2009), 141–147.

[30] Michael Liedlgruber and Andreas Uhl. 2011. Computer-aided decision support systems for endoscopy in the gastrointestinal tract: a review. *IEEE reviews in biomedical engineering* 4 (2011), 73–88.

[31] Mathias Lux. 2013. LIRE: open source image retrieval in Java. In *Proc. of ACM MM*. 843–846.

[32] Mathias Lux and Oge Marques. 2013. Visual Information Retrieval using Java and LIRE. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 5, 1 (2013), 1–112.

[33] Shawn Mallery and Jacques Van Dam. 2000. Advances in diagnostic and therapeutic endoscopy. *Medical Clinics of North America* 84, 5 (2000), 1059–1083.

[34] A.V. Mamonov, I.N. Figueiredo, P.N. Figueiredo, and Y.-H.R. Tsai. 2014. Automated Polyp Detection in Colon Capsule Endoscopy. *IEEE Transactions on Medical Imaging* 33, 7 (2014), 1488–1502.

[35] Tom M Mitchell. 1997. Machine learning. WCB. (1997).

[36] B. Munzer, K. Schoeffmann, and L. Boszormenyi. 2013. Detection of circular content area in endoscopic videos. In *Proc. of CBMS*. 534–536.

[37] B. Munzer, K. Schoeffmann, and L. Boszormenyi. 2013. Improving encoding efficiency of endoscopic videos by using circle detection based border overlays. In *Proc of ICME workshops*. 1–4.

[38] B. Munzer, K. Schoeffmann, and L. Boszormenyi. 2013. Relevance Segmentation of Laparoscopic Videos. In *Proc. of ISM*. 84–91.

[39] Ruwan Nawarathna, JungHwan Oh, Jayantha Muthukudage, Wallapak Tavanapong, Johnny Wong, Piet C De Groen, and Shou Jiang Tang. 2014. Abnormal image detection in endoscopy videos using a filter bank and local binary patterns. *NC* (2014).

[40] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2014. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv:1412.1897* (2014).

[41] Konstantin Pogorelov, Kristin Ranheim Randel, Thomas de Lange, Sigrun Losada Eskeland, Carsten Griwodz, Dag Johansen, Concetto Spampinato, Mario Taschwer, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. 2017. Nerthus: A Bowel Preparation Quality Video Dataset. In *Proc. of MMSYS*.

[42] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. 2017. Kvasir: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. In *Proc. of MMSYS*.

[43] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, and Carsten Griwodz. 2017. ClusterTag: Interactive Visualization, Clustering and Tagging Tool for Big Image Collections. In *Proc. of ICMR*.

[44] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Peter Thelin Schmidt, Carsten Griwodz, Dag Johansen, Sigrun L. Eskeland, and Thomas de Lange. 2016. GPU-accelerated Real-time Gastrointestinal Diseases Detection. In *Proc. of CBMS*. 185–190.

[45] Michael Riegler, Martha Larson, Mathias Lux, and Christoph Kofler. 2014. How 'How' Reflects What's What: Content-based Exploitation of How Users Frame Social Images. In *Proc. of MM*. 397–406.

[46] Michael Riegler, Konstantin Pogorelov, Sigrun Losada Eskeland, Peter Thelin Schmidt, Zeno Albisser, Dag Johansen, Carsten Griwodz, Pål Halvorsen, and Thomas de Lange. 2017. From Annotation to Computer Aided Diagnosis: Detailed Evaluation of a Medical Multimedia System. *Transactions on Multimedia Computing, Communications and Applications* 9, 4 (2017).

[47] Michael Riegler, Konstantin Pogorelov, Pỳl Halvorsen, Thomas de Lange, Carsten Griwodz, Peter Thelin Schmidt, Sigrun Losada Eskeland, and Dag Johansen. 2016. EIR - Efficient Computer Aided Diagnosis Framework for Gastrointestinal endoscopies. In *Proc. of CBMI*. 1–6.

[48] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117.

[49] Jingkuan Song. 2013. Effective hashing for large-scale multimedia search. In *Proc. of SIGMOD/PODS Ph. D. symposium*. 55–60.

[50] Donald F Specht. 1990. Probabilistic neural networks. *Neural Networks* 3, 1 (1990), 109–118.

[51] Håkon Kvale Stensland, Vamsidhar Reddy Gaddam, Marius Tennøe, Espen Helgedagsrud, Mikkel Næss, Henrik Kjus Alstad, Asgeir Mortensen, Ragnar Langseth, Sigurd Ljødal, Ostein Landsverk, Carsten Griwodz, Pål Halvorsen, Magnus Stenhaug, and Dag Johansen. 2014. Bagadus: An Integrated Real-time System for Soccer Analytics. *Transactions on Multimedia Computing, Communications and Applications* 10, 1s, Article 14 (Jan. 2014), 21 pages.

[52] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. *arXiv:1512.00567* (2015).

[53] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. 2016. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on Medical Imaging* 35, 2 (2016), 630–644.

[54] Taffee T Tanimoto. 1958. elementary mathematical theory of classification and prediction. (1958).

[55] The New York Times. 2013. The $2.7 Trillion Medical Bill. (2013). http://goo.gl/CuFyFJ Accessed: 2015-11-29.

[56] Brian Van Essen, Chris Macaraeg, Maya Gokhale, and Ryan Prenger. 2012. Accelerating a random forest classifier: Multi-core, GP-GPU, or FPGA?. In *Proc. of FCCM*. 232–239.

[57] L. von Karsa, J. Patnick, and N. Segnan. 2012. European guidelines for quality assurance in colorectal cancer screening and diagnosis. First Edition–Executive summary. *Endoscopy* 44, S 03 (2012), SE1–SE8.

[58] Yi Wang, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C de Groen. 2011. Computer-aided detection of retroflexion in colonoscopy. In *Proc. of CBMS*. 1–6.

[59] Yi Wang, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C de Groen. 2013. Near Real-Time Retroflexion Detection in Colonoscopy. *BHI* 17, 1 (2013), 143–152.

[60] Yi Wang, Wallapak Tavanapong, Johnson Wong, JungHwan Oh, and Piet C de Groen. 2014. Part-Based Multiderivative Edge Cross-Sectional Profiles for Polyp Detection in Colonoscopy. In *Proc. of BHI*, Vol. 18. 1379–1389.

[61] Yi Wang, Wallapak Tavanapong, Johnny Wong, Jung Hwan Oh, and Piet C De Groen. 2015. Polyp-alert: Near real-time feedback during colonoscopy. *Computer methods and programs in biomedicine* 120, 3 (2015), 164–179.

[62] Yi Wang, Wallapak Tavanapong, Johnny S Wong, JungHwan Oh, and Piet C de Groen. 2010. Detection of quality visualization of appendiceal orifices using local edge cross-section profile features and near pause detection. *BME* 57, 3 (2010), 685–695.

[63] Mingda Zhou, Guanqun Bao, Yishuang Geng, B. Alkandari, and Xiaoxi Li. 2014. Polyp detection and radius measurement in small intestine using video capsule endoscopy. In *Proc. of BMEI*. 237–241.